

09/579,670

REMARKS

The present response is filed in response to the Office Action of January 12, 2004 and is presented within two months of the mailing date of January 12, 2004 of the Final Office Action.

The Applicant respectfully requests that the Examiner reconsider and withdraw all grounds for rejection of the claims in light of the following discussions and claim amendments and the allowance of the Application as amended herein, or, if necessary, an Advisory Action.

Claims 1-16 are presently pending in the application.

Claims 1, 3, 5, 9, 11, 13 and 15 have again been rejected under 35 U.S.C. § 102 over U.S. Patent No. 6,205,449 to Rastogi et al. '449 for a SYSTEM AND METHOD FOR PROVIDING HOT SPARE REDUNDANCY AND RECOVERY FOR A VERY LARGE DATABASE MANAGEMENT SYSTEM, hereafter referred to as "Rastogi et al. '449.

Claims 2, 4, 6, 8, 10, 12, 14 and 16 have again been rejected over Rastogi et al. '449 under 35 U.S.C. § 103(a).

The Applicant acknowledges and respectfully traverses the raised obviousness rejection in view of the following remarks.

The following will first review previously presented discussions of both the present invention and the teachings of Rastogi et al. '449, and will then discuss the distinctions of the present invention as claimed over the teachings of Rastogi et al. '449.

First considering the present invention, the present invention and the claims of the present application are directed to system resource for performing system resource operations requested by a client of the system resource, such as a file server performing file transactions requested by a client. According to the present invention and the claims, the system resource includes a state machine logging mechanism for extracting, storing and restoring state machine information representing the state of operation of the system resource and of the system resource operations, such as file transactions.

3/12/04 - 10:21 AM

09/579,670

In a first aspect of the present invention, the claims recite and define a system resource, such as a file server, with a state machine logging mechanism as including a system resource sub-system, such as a storage sub-system, and a control/processing sub-system that includes a resource control processor performing system resource operations in response to client requests and controlling operations of the system resource sub-system. The control/processing sub-system may be, for example, a file system processor performing file transaction operations in response to client requests and controlling file storage operations of the storage sub-system, and the state machine logging mechanism.

The state machine logging mechanism, in turn, includes a state machine log generator for extracting state machine information defining a current state machine representing a current state of execution of a system resource operation and a state machine log mechanism for storing a sequence of one or more state machines representing the sequential, detailed, state machine level operations of the system in executing the requested transactions. The state machine log generator is responsive to the restoration of operation of the system resource after a failure of the system resource for reading the state machine information from the state machine log mechanism and restoring the state of execution of a current system resource operation.

As described in the specification and as recited in the claims, the state machine log mechanism includes a state machine log mirroring mechanism operating separately from the control/processing sub-system and communicating with the state machine log generator through a local high speed data link for receiving and storing mirror copies of the state machine information. By operating separately from the control/processing sub-system, the state machine log mirroring mechanism is capable of remaining in operation even upon failure of the control/processing sub-system. The state machine log mirroring mechanism is responsive to the restoration of operation of the file server after a failure of file server operations for reading the state machine information from the state machine log mirroring mechanism and restoring the state of execution of a file transaction.

3/11/04 4:22 AM

09/579,670

Also according to the present invention, and as described and defined in the specification, "state machine information" is defined and described as and by the control and data values residing in the machine during a given state, or point in time.

A "state machine", in turn, is described and defined in the specification as a machine or system, such as a system resource or a file server, that executes operations as a sequence of discrete operating "states" wherein, as discussed, a "state" is defined and described by the control and data values residing in the machine during that state, or point in time. The present and next operating state of a state machine are described and defined by the current state of the machine, the state functions of the machine itself, that is, the logic and circuit functions implemented in the machine that determine the responses or changes in state of the machine as a result of a current operating state and any state effecting inputs, and any inputs that will effect the next state of the state machine.

A "state machine", such as a system, sub-system or logical or functional element of a system or sub-system of any form, may therefore may be defined and described as a sequence of one or more state machines wherein each state machine in the sequence of state machines is defined by the current state, that is, control and data values residing in the machine, and the state functions of the machine, that is, the functions or operations that will be executed by the state machine to result in the next state machine. It will be apparent that, for a given system, the state functions of the state machines describing and defining the system, are fixed and implicitly known and need not be specified for each state machine individually. As such, the current state of operation of a system may be defined by the state of the current state machine, that is, the control and data values residing in the state machine, and a sequence of operations executed by the given system may be defined and described by a corresponding sequence of state machines defining the system.

At this point, it must be noted that the state machine logging mechanism of the present invention extracts and stores "state machine information" defining one or more "state machines"

3/11/04 4:20 AM

09/579,670

during the execution a resource operation or transaction rather than only at the conclusion of the operation or transaction, as is typical in transaction logging type systems.

Stated simply, therefore, the state machine logging mechanism of the present invention thereby provides greater "granularity" in recording the execution of an operation or transaction in capturing and storing of a sequence of state machines reflecting the sequential operations of the system during the execution of a requested transaction, so that each transaction is represented and recorded as a sequence of state machines.

To emphasize this aspect of the present invention, it must again be noted and understood that the state machine logging mechanism of the present invention extracts and saves such state machine information during the performance or execution of a resource operation as transaction as a sequence of state machines for each transaction.

Again, each state machine is comprised of the state machine information at each of a sequence of points in the performance or execution of the operation or transaction information.

The state machine logging mechanism of the present invention thereby allows the restoration of a resource server or transaction at a point during the performance or execution of the operation or transaction.

The state machine logging mechanism of the present invention thereby also provides greater assurance that a given resource server operation or transaction is captured for subsequent restoration, if needed, because the necessary information is captured at each stage in the execution of the transaction as a sequence of points in the execution of the operation or transaction, rather than only when the operation or transaction is completed. Stated another way, each change in the system during the execution of a transaction defines a new state of the machine, which is captured and stored as the next state machine in the sequence of state machines defining the transaction. As a consequence, no essential step during the execution of a transaction is or can be lost, thereby allowing reliable reconstruction or restoration of any transaction.

3/12/04 - 02:00 AM

09/579,670

Lastly, it must be noted that the invention as described in the specification and as recited in certain of the claims is directed to an implementation of the present invention in a system resource having a single control/processing sub-system having an associated state machine log generator and a state machine log and is recited as such in claims 1, 3, 5, 7, 9, 11, 13 and 15. The present invention as recited in the claims is, however, also directed to an implementation of the present invention wherein the state machine logging mechanism further includes a state machine log mirroring mechanism that is operationally separate and independent from the control/processing sub-system, such as in claims 2, 4, 6, 8, 10, 12, 14 and 16. That is, in the recitations of claims 2, 4, 6, 8, 10, 12, 14 and 16, the state machine logging mechanism has a state machine log situated in or in association with the control/processing sub-system and a separate state machine log mirroring mechanism that is effectively external and independent of the control/processing sub-system. In this regard, it must be recognized that the separate state machine log mirroring mechanism is not independent from the control/processing sub-system as regards the state machine log mirroring functions, but is separate and independent from the control/processing sub-system as regards basic support functions, such as power to the separate state machine log mirroring mechanism. As such, and for this reason, the separate state machine log mirroring mechanism can continue operation if there is a failure of the control/processing sub-system.

Further in this regard, it must be noted that the state machine log mirroring mechanism is not a separate system analogous to the control/processing sub-system and is not a complete, self-contained and full function system in the manner of a general purpose computer. Instead, the state machine log mirroring mechanism is a specific purpose mechanism designed and constructed only to perform the state machine log mirroring function of capturing and storing a copy of the state machine information and state machines that are stored in the state machine log associate with the state machine log generator that is associated directly with the control/processor sub-system. As such, it will be clear to those of ordinary skill in the arts that the separate state machine log mirroring mechanism does not and cannot operate as a "back-

01104-420 AM

09/579,670

up" unit for the control/processing sub-system in processing requests for operations as the state machine log mirroring mechanism simply does not have the functionality for this type of operation but instead operates solely as a specific purpose mechanism designed and constructed only to perform the state machine log mirroring function of capturing and storing a copy of the state machines comprising the control/processing sub-system while it is processing transactions.

It must also be noted that the claims are also directed to implementations of the present invention having dual control/processing sub-systems with associated state machine logging mechanisms and state machine log mirroring mechanisms. In these implementations, each control/processor sub-system has an associated state machine logging mechanism for its own transactions or operations and a state machine log mirroring mechanism associated with the other control/processing sub-system. As such, each control/processing sub-system maintains its own state machine logging mechanism and maintains a state machine log mirroring mechanism for the other control/processing sub-system and processes transactions independently of the other control/processing sub-system.

In this regard, it must also be noted that in the system resources or file servers having dual control/processing sub-systems, the two control/processing sub-systems operate concurrently, independently and in parallel with each other and with each control/processing sub-system performing its own system resource operations or transactions, and with each control/processing sub-system providing only a residence and support for the state machine log mirroring mechanism of the other control/processing sub-system.

As such, and as described in the specification and recited in the claims, neither control/processing sub-system provides "back-up" for the other control/processing system in the sense of standing by ready to perform or execute transactions or operations for the other control/processing sub-system if the other control/processing sub-system should fail. It must also be noted that because of the concurrent, independent and parallel operation of each

3/11/04 8:20 AM

09/579,670

control/processing sub-system, with each control/processing sub-system executing only the requests for transactions or operations that are directed to it.

At this point it should be noted that in conventional "back-up" systems, such as that taught by Rastogi et al. '449, there typically two identical systems arranged in parallel with one unit being designated as the primary unit and the other as the "back-up" unit. The primary unit typically performs all operations requested of the system while the "back-up" unit is typically idle until it is required to replace the other unit in performing the operations, so that only one half of the total power and resources of the system are available at any time. The system of the present invention, however, makes significantly more of the potential control/processing sub-system capability available to clients of the system because both halves of the system are fully operable

In the systems of the present invention, however, and in fundamental contrast from the typical form of "back-up" systems as taught by, for example, Rastogi et al. '449, the full processing power of the two control/processing sub-systems is available at all times to process requests from clients. If one control/processing sub-system fails, the other control/processing sub-system continues operative to perform the request directed to it and to maintain the state machine information necessary to subsequently restore the failed control-processing sub-system. In addition, the operative one of the control/processing sub-systems may perform requests that would be directed to the other control/processing sub-system if those requests are re-directed to the operative control/processing sub-system. It will therefore be apparent to those of ordinary skill in the arts that only in the event of a failure in one of the control/processing sub-systems of the present invention is the total transaction processing capability of the system reduced to that levels provided by the normal operation of the "back-up" systems of the prior art, such as taught by Rastogi et al. '449.

Next considering the teachings of Rastogi et al. '449. Rastogi et al. '449 describes a system having "hot spare" support wherein the system includes two identical computer systems communicating through a network and wherein, at any given time, one of the computer systems

3/12/04 - 10:01 AM

09/579,670

is designated as the "primary" computer system and the other is designated as the "secondary" computer system. The computer system that is designated as the primary computer system performs all operations, that is, all transactions, of the system and generates a log of all such transactions. The computer system that is designated as the secondary computer system does not perform any operations while the primary system is in operation, but is instead a "back-up" system that stores a copy of the transaction log of the primary system. If there is a failure in the current primary computer system, the secondary computer system becomes the primary computer system and assumes execution of all transactions directed to the system, starting with the copy of the primary system transaction log stored in the secondary computer system, which then becomes the primary computer system.

The primary and essentially the sole object of storing a copy of the current primary system transaction log in the secondary system is to maintain the primary and secondary systems in "synchronization" so that the secondary computer system can assume execution of the transactions directed to the system with minimum loss of the preceding transactions that have been executed by the computer system that was previously the primary computer system.

Rastogi et al. '449 states that, for this purpose, the primary and secondary computer systems each stores a record of the "state" of the computer systems. It will be readily recognized by those of ordinary skill in the relevant arts, however, that Rastogi et al. '449's use of the term "state" differs fundamentally from the term "state" as used in the specification and claims of the present Application.

In particular, in Rastogi et al. '449 the term "state" refers to bits of information stored in the primary and secondary systems that indicate, in each system, whether the system is the current primary system or is the current secondary system and whether the two computer systems are in "synchronization", that is, have matching copies of the primary computer system transaction log. At any given time only one computer system can be the current primary computer system and can execute the system transactions. The secondary system will always be idle as regards the execution of transactions and will function solely to store a copy of the

3/12/04 - 10:01 AM

09/579,670

primary computer system transaction log while waiting to assume execution of the transactions upon failure of the current primary computer system.

It will therefore be apparent to those of ordinary skill in the arts that in the teachings of Rastogi et al. '449 the term "state" has no relationship or meaning with regard to the state of operation of a system at each step in executing a transaction, or even to the actual execution of transactions, but instead relates only to the overall current functional assignments of the two systems and, in particular, to which one is assigned to execute transactions and which one is assigned to store a copy of the transaction log, and not how the transactions are executed.

As taught by Rastogi et al. '449, the primary system transmits a copy of the primary system transaction records to the secondary system through a network connection in either of two circumstances, depending upon the specific designed operation of the Rastogi et al. '449 system. In one circumstance, the primary system transaction log is transmitted to the secondary each time the transaction log in the primary system is "flushed" to disk, that is, is moved from the primary system memory space to the primary system mass storage device for long term storage. In the second circumstance, the primary system will flush its local copy of the transaction log to its own disk when the primary system has transmitted a copy of the transaction log to the secondary system and has received an acknowledgment of the transmission from the secondary system.

It is, therefore, apparent that the present invention is distinguished over and from the Rastogi et al. '449 system for a number of fundamental reasons, which are recited in the claims as amended herein.

For example, and in complete and fundamental contrast from Rastogi et al. '449, the system of the present invention captures and stores state machines representing the detailed operation of the system in the execution of each transaction and stores these state machines while the corresponding transaction is being executed, so that the execution of a transaction may be continued or resumed at any time, not just at the end or beginning of a transaction.

3/12/04 10:01 AM

09/579,670

In order to understand and appreciate this distinction, it must be understood that, as completely described in the present application, a "state machine" as the term is used in the sense of a "state machine" system is a machine or system, such as a system resource or a file server, that executes operations as a sequence of discrete operating "states" wherein, as discussed, a "state" is defined and described by the control and data values residing in the machine during that state, or point in time. The present and next operating state of a state machine system are described and defined by the current state of the machine, the state functions of the machine itself, that is, the logic and circuit functions implemented in the machine that determine the responses or changes in state of the machine as a result of a current operating state and any state inputs, including data, that will effect the next state of the state machine.

A "state machine" system is thereby defined as and understood to be a sequence of one or more "state machines" wherein each "state machine" in the sequence of state machines is defined by the current state of the state machine system, that is, by the control and data values residing in the machine, and the state functions of the machine, that is, the functions or operations that will be executed by the state machine to result in the next state machine. It will be apparent that, for a given system, the state functions of the state machines describing and defining the system, are fixed and implicitly and explicitly known and need not be specified for each state machine individually. As such, the current state of operation of a system may be defined by the "state" of the current state machine, that is, the control and data values residing in the state machine, and a sequence of operations executed by the given system may be defined and described by a corresponding sequence of state machines defining the state machine system as it executes the sequence of operations.

According to the present invention, therefore, a given transaction is represented by the sequence of one or more state machines comprising and defining the state machine system, that is, the control/processing sub-system, during the execution of the transaction. Further according to the present invention, each state machine of the sequence of one or more state

3/12/04 - 10:51 AM

09/579,670

machines is defined by the data and control values residing in the state machine system during each state machine of the sequence. It is this state information defining each of the state machines representing the execution of a transaction that is stored in the state machine log and state machine log mirroring mechanism of the present invention.

In summary, therefore, and according to the present invention, a control/processing subsystem is a state machine system operating to execute a requested transaction and, during the execution of the transaction, is a transaction defined state machine system comprised of a sequence of one or more state machines wherein each state machine of the sequence is represented and defined by the data and control values residing in the state machine system during the state machine period.

Referring again to Rastogi et al. '449, it will be apparent to those of ordinary skill in the arts that the present invention is fundamentally distinguished over and from the system taught by Rastogi et al. '449 in essentially every respect. For example, the transaction log of the Rastogi et al. '449 system captures or generates only transaction records only when the , that is, a record of a completed transaction at the conclusion of the transaction, and actually records, or stores, a transaction record only when the record is transferred into mass storage. In a like manner, the copy of a transaction record is transmitted to and stored in the secondary computer system only when the record is transferred to the primary system mass storage.

In fundamental contrast from the present invention, the Rastogi et al. '449 system captures or generates only transaction records, that is, the data comprising a record of a completed transaction, which has no relationship to a state machine representing the control and data values present in a state machine system and representing the operating state of a system of operation at a given time.

To explain in greater detail, it must first be noted that Rastogi et al. '449 describes a transaction as a sequence of operations at one or more levels and describes a transaction log as storing such transactions. A close reading of Rastogi et al. '449 reveals, however, that Rastogi et al. '449 speaks only of the operations of a transaction, and not the states of

3/12/04 10:01 AM

09/579,670

operation of a system while executing a transaction. In this regard, it must be noted that the only mention of "state" by Rastogi et al. '449 is, as discussed above, stored information indicating which is the primary system and which is the secondary system and whether the primary and secondary systems are synchronized, which is completely unrelated to the actual execution of transactions and even the actual logging of transactions.

In addition, Rastogi et al. '449 essentially only describes representing a transaction in the primary system transaction log in terms of what are in fact the instructions or commands initiating each operation at each level, and perhaps any data input to the operation. At no point does Rastogi et al. '449 describe or even mention a "state machine system", a "state machine", system "state" as represented by the control and data values residing in the system at a given point during the execution of a transaction and defining the "state machine" of the system at that point.

The only reasonable conclusion that could be reached by one of ordinary skill in the arts is therefore that Rastogi et al. '449 is, as stated, referring to a transaction log as containing only the instructions or commands initiating each operation of a transaction, and any input data. The fundamental difference and distinction between what Rastogi et al. '449 stores in the transaction log and what is the state machines that the present invention stores in the state machine log, however, will be apparent from the above discussions of the present invention.

To illustrate this distinction by an example, it must be noted that a step in the execution of a transaction is often effected by the outcome of a preceeding step in the execution of the transaction and that, in most systems, any such information from a preceeding step will be appropriately stored in the processing unit. A conventional system such as that taught by Rastogi et al. '449, however, may well miss such information from a preceeding step as the log stores only the instruction or command for any step and any input data for that step. A state machine system of the present invention, however, because the state machine log stores, at each step in the execution of a transaction, a state machine representing the state of the system during that step, that is, the control and data values present in the system at that time

3/12/04 - 10:51 AM

09/579,670

and effecting the execution of the step, will not miss information from a preceeding step that is required to correctly execute the current step.

The above discussed characteristics of the Rastogi et al. '449 system, that is, the logging of transactions rather than state machines and the logging of a transaction only when the primary system flushes its transaction log to local mass storage, results in yet other fundamental distinctions between the present invention and the teachings of Rastogi et al. '449.

For example, because the Rastogi et al. '449 system captures transactions, that is, the instructions and commands initiating the operations taking place in a transaction, rather than a sequence of state machines, the Rastogi et al. '449 system is essentially limited to either executing each operation of an aborted transaction from the start of each operation or of executing those same operations from the end, that is, in reverse, to "undo" the corresponding transaction. It is for these reasons that the Rastogi et al. '449 system contains both a "redo" log, so that a transaction can be reexecuted from the start, and an "undo" log, so that a transaction can be "undone", or canceled, by "undoing" the transaction from the end.

This conclusion regarding the Rastogi et al. '449 system, and the resulting distinctions between the present invention and the teachings of Rastogi et al. '449, is further supported by the fact that the Rastogi et al. '449 system "logs" a transaction in the back-up log in the secondary systems only when the primary system flushes its transaction log to local mass storage. In other words, the "log" of a given transaction is stored solely in the volatile memory of the primary system until the log is flushed to the primary system mass storage, and is at the same time "flushed" to the secondary system to be stored in the secondary system mass storage.

While Rastogi et al. '449 does not address the implications of these log on flush operations explicitly, it is clear that a "flush" of a transaction from memory to mass storage will typically and normally take place only at the conclusion of the transaction as this will thereby require the storage of only the minimum amount of information about the transaction in order to allow the transaction to be "redone" or "undone". It is noted, in this regard, that Rastogi et

3/12/04 - 10:01 AM

09/579,670

al. '449 does not in fact address whether a log entry contains information taken during the execution of the transaction, or only the starting data, operations commands and instructions and transaction results that are essential to reconstruction of the transaction. It would seem reasonable, however, to those of skill in the arts, to store no more transaction information than is necessary to redo or undo the transaction, which would require only the starting data, operations commands and instructions and transaction results.

It must also be noted that the conclusion that Rastigo teaches only the storage of transactions and not of a sequence of state machines representing a transaction is further supported by the fact that Rastogi et al. '449 speaks only of "redoing" and "undoing" a transaction, which requires only the transaction starting data, operations commands and instructions and transaction results. Rastogi et al. '449 does not even mention restoring or resuming a transaction in the at any point during the actual execution of the transaction, that is, at some point in the middle of the execution of the transaction, which would require the capture and storage of the sequence of state machines defining the transaction.

As such, it is clear that Rastogi et al. '449 does not and cannot capture or store state information of any sort, including state machines, state machine data or the data and control values residing in the control/processing sub-system state machine at each step in the execution of a transaction. It is also clear to those of ordinary skill in the art that this distinction is so fundamental and so basic to the entire design and operation of the two systems as to render the teachings of Rastogi et al. '449 irrelevant with respect to the present invention.

Lastly in this regard, it is noted that the Examiner states that Rastogi et al. '449 teaches the use of state information because the secondary system is available "immediately" to take over the functions of the primary system upon failure of the primary system. It must be noted, however, that not only does Rastogi et al. '449 not even mention state, except with regard to which processor is the primary and secondary processor and whether they are synchronized, that the use of the term "immediately" as used in Rastogi et al. '449 must be read and interpreted in light of the remainder of the teachings in Rastogi et al. '449.

3/12/04 - 10:01 AM

09/579,670

That is, and for example, Rastogi et al. '449 clearly teaches that the logging of a transaction in the back-up log in the secondary systems occurs only when the primary system flushes its transaction log to local mass storage, which is typically only at the end of the transaction. This thereby indirectly states that not only will an "unlogged" or "unflushed" transaction be lost if there is a power failure to the primary processor before the log is flushed, but that since the logging takes place only at a "transaction rate", the term "immediately" does not imply any great speed of response but only that it will take place at the normal paces of transaction processing. This position is further supported when it is noted that the primary system and the secondary system in which the transactions are logged are interconnected through a network so that any restoration of transaction data must take place at network transaction speeds, which is normally a rather slow "immediately". A more reasonable interpretation of the Rastogi et al. '449 teachings is that the secondary system is ready to assume the transaction processing operations of the primary system immediately upon being informed of a failure of the primary system, which is a very different matter and operation from the transaction logging operations.

To consider this distinction further, and in contrast from the Rastogi et al. '449 system, the log mechanism of the present invention is local to the transaction control/processing subsystem for communications purposes and the log mirroring mechanism communicates with the log mechanism through a local, high speed datalink dedicated to that purpose. It is therefore in fact the system of the present invention, and not the system of Rastogi et al. '449, in which logged transaction information, that is, transaction state machines, can be recovered "immediately" from the log mechanism or log mirroring mechanism. It is also only in the system of the present invention, and not the system of Rastogi et al. '449, that can gain benefit of the "immediate" restoration of a transaction record from the log mechanism or log mirroring mechanism because it is only the system of the present that stores "state machines" that enable the restoration of a transaction at any point during the transaction.

3/12/04 - 10:01 AM

09/579,670

It is therefore clear, and the belief and position of the Applicant, that the present invention is completely and fundamentally distinguished over and from the teachings of Rastogi et al. '449 under the requirements and provisions of 35 U.S.C. 102 and 35 U.S.C. 103 because Rastogi et al. '449 teaches and suggests only the capture and logging of transaction records representing the basic transaction operation commands, instructions and data and because Rastogi et al. '449 does not reach or even suggest in any way the capture, logging and restoration of transactions as sequences of state machines including the control and data values residing in and controlling the system at each step of the execution of a transaction.

It is further noted that the Examiner has in fact agreed and stated that Rastogi et al. '449 does not and cannot teach or even suggest the capture, logging and restoration of transactions as sequences of state machines including the control and data values residing in and controlling the system at each step of the execution of a transaction.

It is thus the belief and position of the Applicant that this distinction in itself is so fundamental and basic as to comprise a complete distinction of the present invention over and from the teachings and suggestions of Rastogi et al. '449 under the requirements of 35 U.S.C. 102 and 35 U.S.C. 103.

Finally, and in further distinction between the present invention and the teachings of Rastogi et al. '449, it must be noted that the primary and secondary computer systems of the Rastogi et al. '449 system do not correspond either structurally or functionally with the dual control/processing sub-systems of the present invention. That is, the two computer systems of the Rastogi et al. '449 system are not parallel, cooperating sub-systems within a system, but are completely and separate computer systems.

In still further fundamental distinction between the present invention and Rastogi et al. '449, it must be noted that the dual control/processing sub-systems of the present invention operate independently but concurrently with each control/processing sub-system executing only the requests for transactions or operations that are directed to it, so that the full processing power of the two control/processing sub-systems is typically available at all times to process

3/12/04 4:01 AM

09/579,670

requests from clients. In basic contrast from the present invention, the two computer systems of the Rastogi et al. '449 system do not operate concurrently at any time. Instead, the primary processor alone operates to execute all transactions of the system while the secondary processor is always idle with respect to the execution of transactions, so that only one half the potential processing power of the Rastogi et al. '449 system is available at any time.

In addition, the primary and secondary computer systems of the Rastogi et al. '449 system do not correspond either structurally or functionally with the state machine logging mechanism and state machine log mirroring mechanism of the present invention as each of the primary and secondary computer systems of the Rastogi et al. '449 system is a full function, general purpose computer capable of performing both transaction operations and transaction logging. In contrast, the state machine logging mechanism and state machine log mirroring mechanism of the present invention are both dedicated purpose, specialized function mechanisms that are structurally and functionally different from one another and are directed to separate and distinctly different functions. In a like manner, the control/processing sub-system and a corresponding state machine log mirroring mechanism are structurally and functionally distinguished from the primary and secondary computer systems of the Rastogi et al. '449 system for the same reasons.

It must also be noted that a control/processing sub-system and its associated state machine logging mechanism with the associated state machine log mirroring mechanism cannot be compared, structurally or functionally, with the primary and secondary computer systems of the Rastogi et al. '449 system because the primary and secondary computer systems of the Rastogi et al. '449 system are in fact identical but completely separate and independent systems from one another. In contrast, the state machine log mirroring mechanism is functionally an integral element of the corresponding state machine logging mechanism, even though the state machine log mirroring system resides separately from the state machine logging mechanism as regards such support functions as the power supplies so

3/12/04 - 10:01 AM

09/579,670

as not be involved in a failure of the corresponding control/processing sub-system with which the state machine log generator and log reside.

It is the belief and position of the Applicant that for the reasons discussed above the teachings of Rastogi et al. '449 do not and cannot describe or suggest the present invention as recited in the claims under either or both of 35 U.S.C. § 102 or 35 U.S.C. § 103.

In this regard, it will be noted that the Applicant has considered the Examiner's remarks and in response and in accordance with the above arguments has amended the claims herein above to more explicitly and clearly recited these fundamental distinctions between the present invention and the prior art cited by the Examiner. These amendments, however, do not add any new matter to the present application or to the claims, and do not alter the subject matter or scope of the claims.

The Applicant, therefore, respectfully requests that the Examiner reconsider and withdraw all rejections of the claims as amended herein under either or both of 35 U.S.C. § 102 or 35 U.S.C. § 103 over Rastogi et al. '449, and the allowance of the claims as amended herein.

If any further amendment to this application is believed necessary to advance prosecution and place this case in allowable form, the Examiner is courteously solicited to contact the undersigned representative of the Applicant to discuss the same.

In view of the foregoing, it is respectfully submitted that the raised rejection(s) should be withdrawn and this application is now placed in a condition for allowance. Action to that end, in the form of an early Notice of Allowance, is courteously solicited by the Applicant at this time.


The Applicant respectfully requests that any outstanding objection(s) or requirement(s), as to the form of this application, be held in abeyance until allowable subject matter is indicated for this case.

3/12/04 - 10:01 AM

09/579,670


In the event that there are any fee deficiencies or additional fees are payable, please charge the same or credit any overpayment to our Deposit Account (Account No. 04-0213).

Respectfully submitted,


Gary D. Clapp, Reg. No. 29,055
Customer No. 020210
Davis & Bujold, P.L.L.C.
Fourth Floor
500 North Commercial Street
Manchester NH 03101-1151
Telephone 603-624-9220
Facsimile 603-624-9229
E-mail: patent@davisandbujold.com

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being transmitted via facsimile to the United States Patent and Trademark Office on: March 12, 2004.



Gary D. Clapp
Type name of person signing certificate

3/12/04 - 10:51 AM